CLAIMS

What is claim is:

1. A method of securely managing a number of data files in non-volatile storage in order to restore data after abortion of a write operation, the data being stored in a record oriented data structure with each of the records containing a status byte in addition to the data contents, wherein, at all times during the write operation, all of the data files affected by the write operation contain either all of the data stored prior to the write operation, or all of the data as modified subsequent to the write operation.

2. A method as claimed in claim 1, where the or each file affected by a write operation comprises a plurality of records, one of said records in the or each file containing the data prior to said write operation and another of said records being modified according to said write operation, each of said records also comprising a status byte indicative of the status of the data contained therein.

3. A method as claimed in claim 2, wherein said data prior to the write operation, in each file, is retained as the active data in the case of a power failure, until all files have been successfully updated according to said write operation.

4. A method as claimed in claim 1, wherein each record contains a synchronization byte, indicating a relationship with records of other files.

5. A method as claimed in claim 3, wherein each record contains a synchronization byte, indicating a relationship with records of other files.

6. A method as claimed in claim 5, each record comprising a first pointer indicating the current data-containing record of a previous file and a further pointer (PTR 3) indicating the current data-containing record of a subsequent file.

7. A method as claimed in claim 6, comprising a second pointer (PTR 2) indicating the current data-containing record of that file.

8. A method of securely managing EEPROM data files in order to restore data after abortion of a write operation, the data being stored in the files in a record-oriented data structure, such that the data in all files affected by the write operation is consistent with respect to completion of the write operation, and wherein information concerning the status and location of the consistent data is stored in the record oriented data structure together with the data.

9. A method according to claim 8, wherein two or more data files are affected by said write operation, and wherein new or modified data is written into said files in a cyclic manner, wherein each file comprises an indication of the number of records contained in said file and a plurality of records, and wherein each record comprises an indication of the status of the data in said record, a synchronisation number synchronising with records of other files, and said data.

10. A method according to claim 9, comprising determining a current active record of a first of said files, and a working record of said first file; setting the synchronization number of the working record of said file to the synchronization number of the current active record; copying the data stored in said current active record into said working record and adding to or modifying said data according to said write operation, in said working record; changing the status of said working record of said file to 'active'; repeating said steps for each further file; and changing the record status of said original current active record of said first file to 'inactive' as an indication that said write operation is complete.

11. A method as claimed in claim 10, wherein said step of determining the current active record and the working record of said files comprises searching for the first record in said file whose status byte indicates 'active' status and setting this record as said current active record, and setting the subsequent record as said working record.

12. A method as claimed in claim 11, comprising:

adding to or modifying the data of a record in the first file by:

identifying the current active record of said file and a working record and copying the data to be added to or modified from the current active record to the working record;

modifying the data in said working record in accordance with the write operation; wherein the status byte of said current active record indicates that that record is 'fully active' and the status of said working record indicates that that record is 'inactive';

setting synchronization indicator pointers to indicate that said file is said first file and to indicate that no further files have been modified;

identifying a current active record and a working record of a second file and copying the data from the current active record to the working record; modifying the data in the working record according to said write operation, wherein the status byte of said active current record indicates that the data in this record is "fully active" and the status byte of the working record indicates that this record is 'inactive';

setting synchronization indicator pointers to indicate the link between this file and said first file, and changing said synchronization indicator pointer of said first file to indicate its link with said second file; and

repeating these steps for said second file for any subsequent files, wherein

for the last file affected by said write operation, after setting said synchronization indicator pointers, determining that this is the last file, setting an indication pointer to indicate that no subsequent files are affected by said writing operation; and

setting the status byte of each of said working records of said affected files to a 'fully active' state, whereupon the write operation is complete and the modified data is the active data in all files.

13. A method as claimed in claim 12, wherein, upon interruption of said write operation at any stage, either all current active records of all files affected by said operation are set as 'fully active' records, and the data contained in said files prior to the start of said write operation is the current active data, or all working records of all files are set to a 'fully active' status, in which case all files contain the modified data due to said write operation as said active data.

14. A method as claimed in claim 13, wherein interruption of said write operation during or immediately after the step of modifying the data in the working record of said first file results in the current active record of said first file remaining as the 'fully active' data record, at which time no further files have been modified and all of the 'active' datable files correspond to the data prior to the write operation.

15. A method as claimed in claim 13, wherein an interruption of said write operation during or subsequent to the setting of the synchronization indicator pointers in said first file results in the current active record of said first file remaining as the 'fully active' data record, at which time no further files have been modified and all of the 'active' datable files correspond to the data prior to the write operation.

16. A method as claimed in claim 13, wherein an interruption of said write operation during or immediately after the step of modifying the data in the second or subsequent files results in the current active record of said second or subsequent file remaining set as said 'fully active' record, and, since said synchronization indicator pointer of said first file still indicates that said current active record is still said 'fully active' record of said first file, the currently active data of both or all of said files remains as that prior to the start of the write operation.

17. A method as claimed in claim 13, wherein an interruption to said write process during or immediately after modifying the data in the working record of the last file affected by said write operation, results in all of the current active records of all of said files being retained as said fully active records, wherein the currently active data corresponds to the data prior to the write operation.

18. A method as claimed in claim 13, wherein an interruption of said write process during or immediately after modification of the data in the working record of the last file affected by said write operation, causes all working records of all of said files to become set to 'fully active' records, such that all files contain data modified as a result of said write operation as the currently active data.

19. A method as claimed in claim 12, wherein, when all of said write steps have been successfully completed, without an interruption, said synchronization indicator pointers are used to indicate the links between the modified records of the files affected, and all working records are set to status 'fully active' and said current active records are set to status 'inactive'.

20. A system for securely managing EEPROM data files so that the data can be restored after abortion of the write operation to said data files, the system comprising an EEPROM, and means for writing data to said EEPROM, said EEPROM comprising a number of data files, each data file comprising a plurality of records in a record oriented data structure.

21. A computer programme arranged such that when it is run on a computer, it executes a secure management of EEPROM data files so that data can be restored after abortion of a write operation, the programme being arranged to execute the steps of claim 1.

22. A computer programme arranged such that when it is run on a computer, it executes a secure management of EEPROM data files so that data can be restored after abortion of a write operation, the programme being arranged to execute the steps of claim 7.

23. A computer programme arranged such that when it is run on a computer, it executes a secure management of EEPROM data files so that data can be restored after abortion of a write operation, the programme being arranged to execute the steps of claim 8.

24. A computer programme arranged such that when it is run on a computer, it executes a secure management of EEPROM data files so that data can be restored after abortion of a write operation, the programme being arranged to execute the steps of claim 13.

25. A computer programme as claimed in claim 21, which is stored on a data carrier medium.